Real Life Grasping using an Under-actuated Robot Hand – Simulation and Experiments

Johan Tegin, Boyko Iliev, Alexander Skoglund, Danica Kragic, Jan Wikander

***Abstract*—We present a system which includes an under-actuated anthropomorphic hand and control algorithms for autonomous grasping of everyday objects. The system comprises a control framework for hybrid force/position control in simulation and reality, a grasp simulator, and an under-actuated robot hand equipped with tactile sensors.**

We start by presenting the robot hand, the simulation environment and the control framework that enable dynamic simulation of an under-actuated robot hand. We continue by presenting simulation results and also discuss and exemplify the use of simulation in relation to autonomous grasping. Finally, we use the very same controller in real world grasping experiments to validate the simulations and to exemplify system capabilities and limitations.

***Index Terms*—Manipulators, Simulation, Robot tactile systems, Modeling**

I. Introduction

ROBUST grasping and manipulation of objects is one of the key research areas in robotics. There has been a significant amount of work reported on how to achieve stable and manipulable grasps [1], [2], [6], [12], [16], [19]. This paper presents a system which includes an under-actuated anthropomorphic hand with control algorithms for autonomous grasping of everyday objects. The system can be used both for simulations as well as real experiments and is intended for development of robotic hands and their control. The main focus of this paper is on improving grasp formation and to ensure a secure grasp. The work can however also be used to choose for example the approach vector or any other parameter in a grasping task. Doing so, the controller and approach vector can be chosen not only based on perceptual cues but also on experience that certain control parameters and approach vectors will result in stable grasps.

The presented framework is considered in a domestic setting where a stable grasp is paramount to manipulation success. The system can accomplish robust grasping by hybrid force/position control of at least a few common shapes that fit the morphology of the hand. We do not consider dextrous manipulation, but focus on more coarse manipulation tasks such as picking up and releasing objects common in home environment. Imperfect vision and poor knowledge of objects are matters of fact under such conditions. Object models and their position and orientation cannot be expected to be completely known, hence the grasp execution must allow for some degree of uncertainty in those respects.

Manuscript received March 23, 2009. This work was supported in part by the Knowledge Foundation through AASS at Örebro University.

J. Tegin and J. Wikander are with the Mechatronics Laboratory, Machine Design, Royal Institute of Technology, Stockholm, Sweden (phone: +46-8-790 95 22; fax: +46-8-20 22 87; e-mail: johant@md.kth.se).

B. Iliev and A. Skoglund are with the Center for Applied Autonomous Sensors Systems, Örebro University, Örebro, Sweden

D. Kragic is with the Computational Vision and Active Perception Laboratory, KTH NADA/CVAP, Stockholm, Sweden

In this article, we describe how the performance of the grasp controller can be evaluated and improved by a combination of dynamic simulations and real experiments. As a result, we can specify how robust the grasp strategy is with respect to perceptual uncertainties in terms of object position. The framework allows for investigations of robustness with respect to other object and control properties as well.

As simulations are shown to correspond well with the results from real life grasping under certain conditions, the simulator can be a valid tool for gaining grasp experience. The choice of the suitable grasp can be based on simulated experience.

The contributions of the work presented here are as follows:

1) We show that the proposed control framework [23] can be applied successfully to under-actuated robots and for real life robot grasping. We do this by implementing part of the robot hand model in a simulation environment and the rest in an existing grasp simulator without support for under-actuated robots and by performing real robot experiments.
2) We show how grasp simulation can be an important tool in establishing the required accuracy and precision for manipulation in domestic environments, and evaluate the quality of different grasp types with respect to inaccuracies in pose estimation. This is an important issue that commonly occurs in robotic systems. The reasons may be that the calibration of the vision system or hand–eye system is not exact or that a detailed model of the object is not available.
3) We benchmark the GraspIt![12] simulation environment and the simulation of a novel robot hand [24] by comparing the simulation results to the performance of the real robot hand in real experiments. This is possible since the very same control implementation can be used to control the simulation and real robot hand by only changing a handful of parameters and changing the target from the simulated to the real robot hand.

II. Related Work

The work on automatic grasp synthesis and planning is relevant to the ideas presented here [13], [14], [17], [19]. But the results in this paper can also be used in other settings, such as in combination with Programming by Demonstration, where the user teaches the robot tasks by demonstrating them [20], [21].

In automatic grasp synthesis, it is commonly assumed that the position, the orientation, and the shape of the object are known [13]. Another common assumption is that it is possible to extract the outer contour of an object and then apply a planar grasp [14]. The work on contact-level grasp synthesis concentrates mainly on finding a fixed number of contact locations without considering the hand design [2], [5]. Considering hand kinematics and a priori knowledge of the feasible grasps has been acknowledged as a more flexible and natural approach toward automatic grasp planning [13], [18].

The method proposed in [13] presents a system for automatic grasp planning for a Barrett Hand [25] by modeling objects as sets of shape primitives; spheres, cylinders, cones, and boxes. Such object models could also be created automatically [9]. Each primitive is associated with a set of rules to generate a set of candidate pre-grasp hand positions and configurations. The ideas are further extended in [4] where a knowledge base for grasping novel objects is created, but without considering control and the dynamics of the grasp formation process. Compared to many grasp planning methods, the heuristic approach is simpler since it does not require searching for grasp points but only an approach vector. The choice of suitable grasps is hence based on object pose and shape and not only to a set of grasping points.

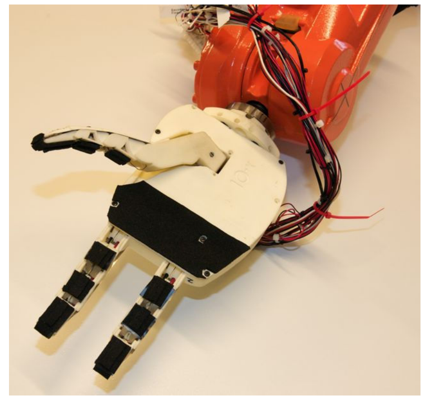
Reaching a pre-computed grasp can be difficult. If the models are not perfect and with an under-actuated end effector, it may be impossible. Grasping under large uncertainties hence requires that we close the force control loop as exemplified in [16]. It is then possible to design controllers that react to the natural changes in object and hand during grasp formation. Another option is to exploit compliance to reduce the need for advanced active control [6].

The work in [27] is highly relevant as it contains detailed analysis, dynamic simulations, and comparison of the two for an under-actuated finger. In comparison, we use a more simplistic robot design and extend control to beyond the finger level. To the best of our knowledge, there are no commercial solutions that allow dynamic simulation of under-actuated robots, a view supported also in [27].

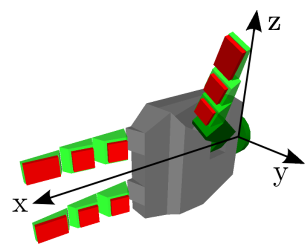
Advanced control and dextrous manipulation has previously been shown in e.g. [15] and using the perhaps most advanced robot hand, the DLR hand [11]. However, we believe that many tasks useful for a service robot acting in a home environment can be performed using more simple hardware by exploiting tactile information, intelligent control, and extensive simulation in the development process. We use an under-actuated robot hand, the KTHand [24]. In many situations, grasping using an under-actuated hand relieves the need for detailed grasp planning as it only requires an approach vector and initial grasp position to succeed. However, it does not offer detailed control and is less capable than a properly controlled fully actuated robot hand.

We wish to point out that we do not consider grasp from a kinetic point of view. We consider the actual grasp result from the reactive hybrid force/position controller and the grasp formation process. As the object will react to contact forces and that these motions of the object affect grasp quality, we are convinced that studying grasp formation is essential to achieve robust and secure grasping.

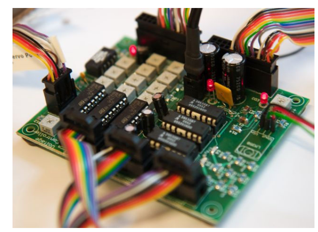
III. Robot Hardware



(a) The KTHand.



(b) The KTHand model.



(c) Interface board including sensor amplifiers, motor drivers, and voltage conversion.

Figure 1: The KTHand, its model, and the electronics interface.

A robotic end effector for service robotics should be affordable, lightweight, equipped with tactile sensors and be capable of performing useful tasks. The KTHand robot hand [24], Figure 1, was designed to fulfill those criteria. It is a three fingered under-actuated hand of anthropomorphic design. The total parts cost for a KTHand is less than €1000. All drawings, the bill of material, and assembly instructions are freely available on the web at *www.md.kth.se/kthand*. The morphology allows many useful grasps, but the sensor configuration needs to be further improved to reach all these grasps using tactile sensors and force control. Even though performance cannot rival that of the DLR hand [11], we believe that the affordable KTHand will enable grasping where a high performance articulate dextrous manipulator like the DLR hand cannot fit within the budget; for certain grasping research, student projects, and affordable service robotics. Its low mass and force controllability makes it applicable where a more industrial like grasper as the Barrett or the Schunk (Dextrous Hand SDH, www.schunk.com) hands may not be appropriate. The KTHand is also simple from a mechanical view making it serviceable, and it is – thanks to the under-actuation – less demanding from a control and planning perspective.

The KTHand has ten degrees of freedom, three for each finger and one for the thumb base, and four degrees of actuation. Each finger is controlled by a direct current motor

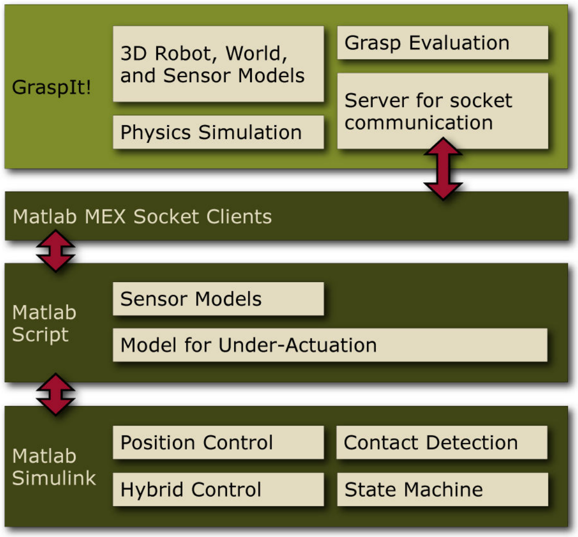


Figure 2: Structure of the simulation software.

which – through a gearbox – rotates a pulley upon which a tendon is wound up. Shortening the tendon flexes the finger. A leaf spring runs along the finger and acts as abductor but is also an integral part of the joint design. The position of the pulley is measured by an magnetic absolute position encoder. Each motor axis is also equipped with a currently not utilized optical encoder.

Each phalanx and the palm is equipped with a tactile force sensor cut to appropriate length from a force sensing resistor no. 408 from Interlink Electronics. It is covered by rubber foam for protection, compliance and increased friction. The sensors cover the finger pads of the distal phalanges, but not yet the fingertip. After calibration the sensors have an accuracy of approximately 10% for similarly shaped objects, but the error is larger as a sharp edge will give a higher force reading than a rounded edge. The palm sensor is not utilized in the current controller, but can be used for grasp force control, approach control, or grasp quality evaluation. A custom electrical interface featuring sensor amplifiers and motor drivers, see Figure 1 c, connects the hand to a rapid control prototyping environment from dSpace (www.dspace.de).

The KTHand is mounted to an ABB IRB140 industrial robot arm. The arm is not intended for use in a home environment, but allow us to position the hand and to manipulate objects in our experiments.

IV. Simulation Framework

The grasp simulator GraspIt! [12] is used for dynamic simulation of the grasp formation process. It allows remote extraction of contact information, setting joint torques, extracting quality measures, advancing simulation, and setting most parameters required to control and monitor the simulation. To avoid socket exhaustion on modern computers, we rewrote the socket client layer to allow socket reuse. Also, we've added some functionality to the server to allow extraction of e.g., grasp quality and saving images. Unfortunately, the dynamics implementation in GraspIt! does not include joint friction. We simulate joint friction by adding a derivative term to the controller. This is the major reason for the short time step required (as small as 0.1 ms in certain cases) and hence the reason for the several minutes required to simulate a grasp formation process.

The controller is implemented in Matlab Simulink to allow for easy rapid control prototyping using Real Time Workshop and dSpace soft- and hardware. Exchanging one simulink block and defining some additional parameters is all that is required to change the target from the simulated to the real hand.

As the KTHhand has four degrees of actuation and 10 degrees of freedom, a model of the under-actuated hand was developed and implemented in Matlab, see Figure 2. This allows realistic simulation in GraspIt! that does not support under-actuated robotics by default.

V. Control

We restrict ourselves to hand control only. That is, we consider the wrist as fixed in space during grasp formation. The hand with its autonomous grasp control can hence be used as a plug-in device in service or industrial robots. To grasp an object, e.g. a bottle, the robot only has to approach the object with an open hand and activate the grasp controller. It should however be noted that compliance control of the wrist and arm can significantly improve grasp formation and facilitate control [6].

A. Grasp Control

Touch sensors capable of detecting the normal force are mounted to the links of each finger of the robot hand. This type of touch sensors are available at a low cost and are easy to mount to an existing robot hand as e.g. the KTHand. Considerations on different tactile sensors are put in to perspective in e.g. [8], [10], and [22].

The hybrid force/position controller uses these tactile sensors to control the grasp force. Position control using position encoders maintain the desired finger configuration and hence object position. Here, we try to reach a "fully closed" grasp, i.e. we try to reach the most stable power grasp where the object is supported also by the palm. We have chosen to focus on the higher level control algorithm although we acknowledge the importance of low-level control design investigated in detail in e.g. [26].

B. Controller Design

To enable a more intuitive formulation of the controller – as opposed to decentralized control of reference trajectories and/or torques in joint space – a linear transform T is used to transform the *finger positions* q to more intuitive control variables $x = Tq$. q_1 is the angle of thumb base rotation. q_2, q_3, and q_4 are the amount of tendon retraction for the "index finger", "ring finger", and thumb respectively. Using a diagonal matrix T will hence allow position control of the tendon length. Using a non-diagonal matrix T enables control of linear combinations of the finger positions q. For example can total closure – defined as the sum of the closing angle for all three fingers – be controlled using forces, positions, or any combination thereof. See [23] for more detail.

The choice of the matrix T is paramount to grasp controller behavior, but quite straight forward. Here, weighted sums of contact forces and finger positions are controlled. For example, using the KTHand, we could use the transform below, Equation 1, where the first row is position control of the thumb base rotation, the second is used to sum the contact forces for all three fingers, the third row is used to express that the two opposing fingers average position should equal $3/2$ of the thumb position, and the fourth row is used to control the difference in position of the two fingers opposed to the thumb. We sum the tactile force readings for all the links of each finger to give on single force value for each finger.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1/3 & 1/3 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}. \tag{1}$$

The controller output u_X (in the transformed space) is computed using a proportional controller $u_x = De$ where D contains controller gains and e is an error vector. The actuator signal for the proportional controller can then be computed as $u_q = T^T f = T^T De$. In this paper, we use one force controller and three position controllers. Thus, the error e is computed using the desired $[des]$ values (in the transformed space) and the actual $[act]$ sensor readings (of position and force) as

$$\begin{aligned} e &= \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}^T \\ e_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} e_x \\ e_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} e_f \\ e_3 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} e_x \\ e_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} e_x \\ e_f &= f_{des} - f_{act} = f_{des} - T f_{act}^{tactilesensors} \\ e_x &= x_{des} - x_{act} = x_{des} - T q_{act}. \end{aligned} \tag{2}$$

C. Control Implementation

The Simulink implementation allows for changing the control laws and control gains easily. The control scheme is implemented as a state machine, with the following states:

1) *Closing:* Position control is used to close the fingers until contact.
2) *Hybrid force/position control:* Engaged when all relevant fingers have contacted the object and slowly ramps the force and position references while softly reducing the influence of the closing controller to zero for a bumpless transfer.
3) *Retraction*: position control where the fingers are retracted completely and the object released.

As the control framework is designed to facilitate implementation of different controllers, multiple controllers can be developed and tested swiftly. The controller for grasping the book in Section VII-E was developed in less than an hour. In addition, thanks to the force control, a single controller can be used for several different objects.

VI. Grasp Quality

Evaluating grasp quality in simulation is possible since all contact points, contact forces and friction are all known. In reality, it is more complicated as these parameters are difficult to measure, especially estimating friction and detecting all contact points.

GraspIt! [12] can calculate the grasp quality which can be automatically retrieved in our environment to facilitate evaluation. The measure we choose to use is the radius of the largest wrench space ball, centered at the origin, which can just fit within the unit grasp wrench space (ϵ_1). Thus, the closer the grasp quality is to one, the more efficient it is. Most of the force closure grasps are sufficient for a secure grasp that allows manipulation. We only rate grasps after the controller has reached its final state. A force closure grasp is considered a successful grasp.

There is a need for a good grasp quality metric using only the information available in a real setting. As the required grasp quality depends upon the task at hand as well as object parameters that can be hard to estimate, it is a delicate task. Looking at the amount of closure for the fingers and the force sensor measurements gives us a crude grasp quality estimate, but only if we already have or can establish good estimates of mass and friction of the object.

VII. Results

This section presents some example uses of the system and some results from simulations as well as real life grasping.

A. Simulation – Estimating the Required Position Accuracy

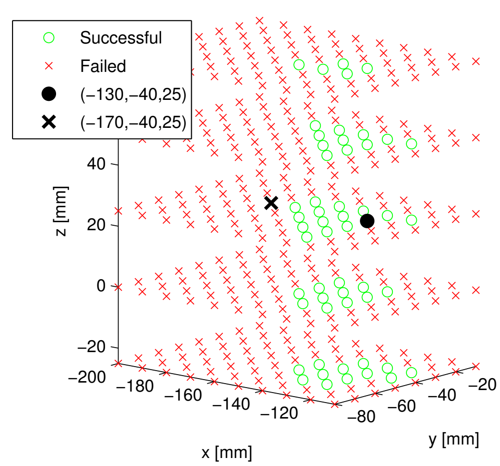
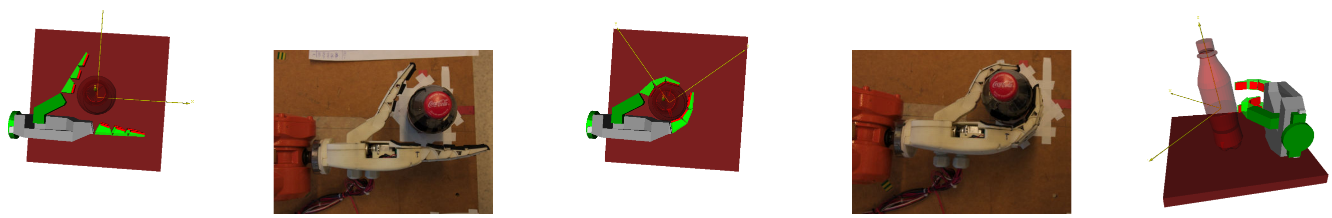


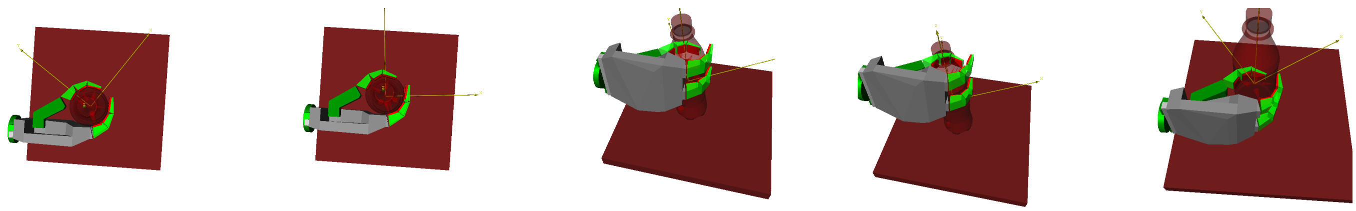
Figure 3: Success is a force closure grasp. The special markers are the initial position for the grasps in Figure 4.

Establishing the required accuracy of object models and pose estimation is important for several reasons. First, a very high accuracy significantly increases the cost of a robotic system. Second, quantifying the required accuracy enables us



(a) Initial grasp position in simulation. (b) Initial grasp position in reality. (c) Final grasp in simulation. (d) Final grasp in reality. (e) A failed grasp.

Figure 4: A successful grasp from $(x, y, z) = (-130, -40, 25)$ and a failed grasp from $(x, y, z) = (-170, -40, 25)$. The positions are marked with a fat black circle and cross respectively in Figure 3.



(a) The highest quality grasp, $\epsilon_1 = 0.0471$. Initial grasp position $(x, y, z) = (-130, -50, 25)$.

(b) The second highest quality grasp, $\epsilon_1 = 0.0463$. Initial grasp position $(x, y, z) = (-120, -30, 25)$.

(c) The worst but yet force closure grasp, $\epsilon_1 = 0.0003$. Initial grasp position $(x, y, z) = (-140, -50, 50)$.

(d) The second worst but yet force closure grasp, $\epsilon_1 = 0.0014$. Initial grasp position $(x, y, z) = (-140, -50, 75)$.

(e) A successful low grasp, $\epsilon_1 = 0.0249$. Initial grasp position $(x, y, z) = (-140, -40, -25)$.

Figure 5: Some example grasps.

to establish the capacity of a certain robot system. Third, if these requirements can be quantified, we can use any margins to relax other system requirements; required model accuracy, sensor precision, controller performance et c.

To investigate the grasping properties of the KTHand with respect to the position where the grasp controller is engaged, i.e. the initial position, we evaluated a three dimensional grid with 80 points in five layers for a total of 400 points in search for force closure grasps, see Figure 3. The desired grasp force is set to 10 N. The results are summarized in Table I. We consider the force closure grasps as successful. The simulation shows that the vertical position of the hand is less important unless the bottle is grasped near its narrow top and that the required position accuracy of the hand with respect to the bottle is approximately 10 mm in the horizontal plane.

Assuming that the model of the manipulator and object both are perfect; we can predict that if the bottle position with respect to the manipulator is known to within 10 mm in the in the horizontal plane and to within 25 mm in the vertical direction, the bottle can be approached without touching it, followed by engaging the controller, and the eventual reach a secure force closure grasp. These results are also coherent with prior simulations using the Barrett Hand to grasp an orange from above that also suggested a required position accuracy in the 10 mm range.

These results can be seen as off-line training for grasping of everyday objects. As certain objects are used many times, it can justify the effort to learn how to handle them in the most reliable way by performing simulations under different conditions.

Table I: Summary for the simulated bottle experiments.

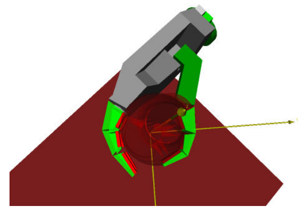
Entity	Value
Number of evaluated configurations	400
No. of configurations w/o collision between object and hand	253
Number of force closure grasps	54
Average grasp quality for successful grasps	0.0168
Standard deviation of grasp quality for successful grasps	0.0114

B. *Experimental – Validating the Required Position Accuracy*

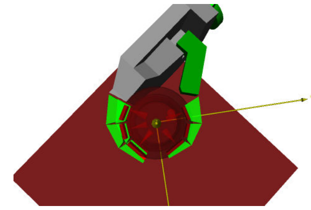
To validate the simulation results in Section VII-A, we tested all force closure grasps from our simulation where the hand grasps close to the center of the object in the vertical direction, $z = 25$, in reality using an ABB IRB140 robot arm and the KTHand. All grasps were force closure also in reality, an example is shown in Figure 4. However, the simulations did not perfectly predict the grasp formation process in all cases. Out of the 15 experimental trials, nine correspond well to the simulations. The remaining six were force closure, but the final position of the bottle was not predicted properly. We believe the reason for this is the friction modelling in the simulation; the friction coefficient for the object overrides that of the object it contacts. Thus, the contacts between the bottle and the hard table have the same friction coefficient as the contacts between the bottle and the rubber foam on the fingers. The higher friction between the robot hand and the bottle in reality will not allow the bottle to slide into the grasp in the same way as in simulation. A lower friction in the fingers may actually give a higher quality grasp under these circumstances as the bottle more easily can slide towards the palm and form closure during grasp formation.

The sensors on the real KTHand does not actually reach the fingertip as opposed to the sensors in simulation that really extends all the way to the fingertip. This was the the second issue we experienced, but with less effect on the results. Thus contacts that occur on the very fingertips are not detected correctly in the experiments. As contact detection currently only involves tactile forces, the switch to the hybrid controller never occurs or occurs too late. The closing controller will however hold the object in a secure grasp, but not with the desired grasp forces.

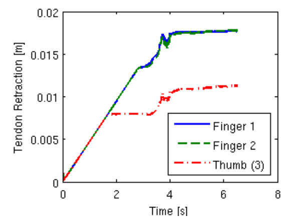
Data from a real experiment is shown in Figure 6. The control loop runs at 1000 Hz, which is significantly slower than in simulation where time steps down to 0.1 ms are used. A dead-band is introduced to the position encoders due to limitations in the dSpace I/O. It is clearly visible as a step in the position signals in Figure 6. The low pass filter for the sensor signal has a time constant of 0.01 s as opposed to 0.1 s in the simulations required to deal with the contact modelling problems in GraspIt! In simulation, the controller output affects tendon force, in reality, motor voltage. The force controller gains are hence somewhat different in simulation and the real experiments. In simulation, a gain of 30 is used. In reality, we use a gain of three.



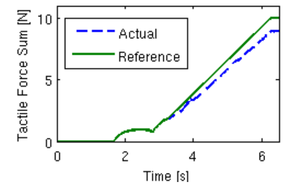
(a) Using a symmetric closing ratio, the fingertips do of the fingers do not wrap the object fully.



(b) Changing the ratio of finger to thumb tendon retraction to $18/12 = 3/2$ results in a much improved grasp.



(c) The tendon retractions over time for the modified controller.



(d) Sum of the tactile forces over time for the modified controller.

Figure 7: Using the simulation environment, we can swiftly derive and implement an improved controller.

C. Simulation – Improving the Controller

Here we show how simulations can be used to facilitate control development within the current control framework. In previous papers, we have primarily used the Barrett Hand [25] and a control algorithm which tries to keep the closure of the thumb and average of the two other fingers equal. Since the morphology of the Barrett Hand is very different from that of the KTHand, this strategy is not ideal, which can be clearly seen in simulation.

The thumb on the KTHand is positioned further back than the two other fingers. Inspection of the final grasp suggests that the tendon for the thumb should be retracted approximately 12 mm and the other two fingers approximately 18 mm for an improved grasp. That is, we wish the fingers to close more than the thumb. The ratio of thumb to finger closure is changed from $\frac{1}{1}$ to $\frac{18}{12}=\frac{3}{2}$. This significantly improves the resulting grasp quality, see Figure 7, and is the control transform in Equation 1.

D. Simulation – Grasping an Orange

The orange is modelled as a sphere with a diameter of 85 mm. For the real experiments, we use a plastic ball of the same size to have good correspondence between model and object. As the orange model is a triangle mesh and resting on only one corner from the start, the smallest numerical perturbation will make it fall over to rest on a triangle instead. To avoid this, we've added a 2 mm square flat area at the bottom of the orange. We believe this is small enough no to affect the objects' reaction to grasping forces significantly. We also tested using a real orange as can be seen in Figure 8.

The controller from the previous experiments grasping a bottle is therefore used also for grasping an orange. Figure 8 shows the results from grasping an orange from above. The orange simulation had to be performed using a very small time step to get realistic results. This is due to numerical perturbations in the contact formulations implementation in GraspIt!. Hence, the execution of the 4 s grasp formation sequence took 493 s on an Intel Core2Duo T9400 processor running Linux 2.6.27. Most simulations can however be performed in shorter time with reasonable accuracy.

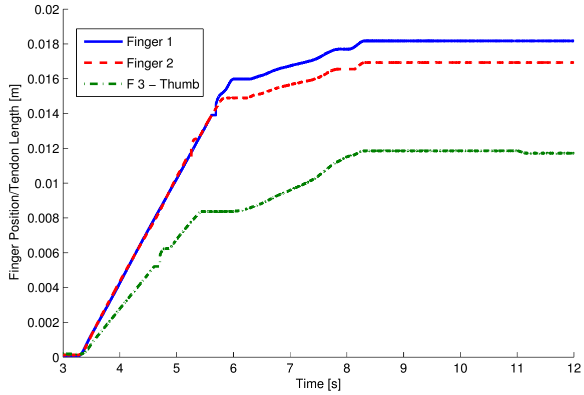
One week later we tried grasping the original orange. As it had shrunk slightly, the sensors never detected any contact. This clearly shows the non-trivial environment a domestic robot must cope with. As the orange was still quite large, one finger and the thumb detected contact, but one finger did not as the contact was on the finger edge only. The controller currently uses tactile information only to detect contact, hence it never switches from closing position control to hybrid force/position control. The orange is pushed to the side by the finger that does not detect contact, see Figure 8 e.

E. Grasping a Book

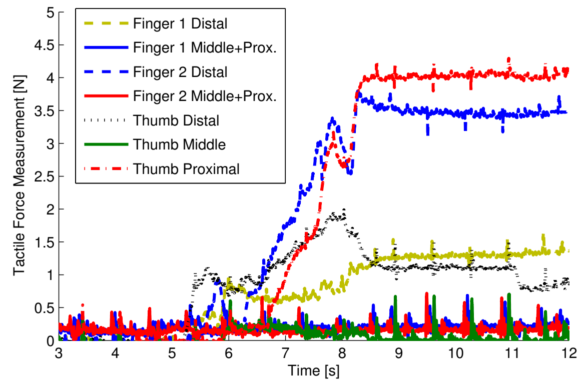
To test the versatility and the use-ability of the system, we decided to simulate and grasp a book. First, we developed the controller in simulation and then transferred it the real robot system. This process took approximately one hour. The book is grasped primarily by thumb motion and required us to modify the closing velocities of the fingers and also the contact detection criteria as the contact between book and thumb was more on the finger tip than towards the finger pad. The results from simulation and real life grasping can be seen in Figure 9.

VIII. Discussion

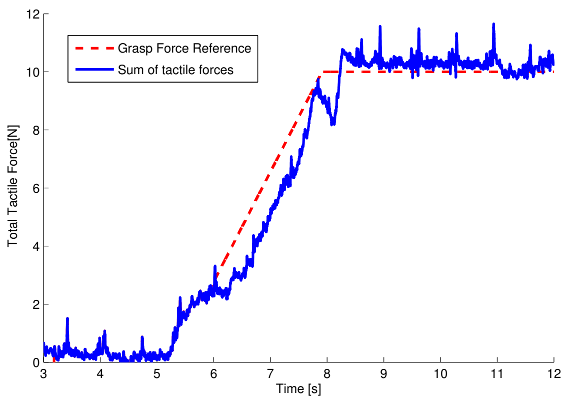
Here we discuss the limitations of the current system and suggest future improvements.



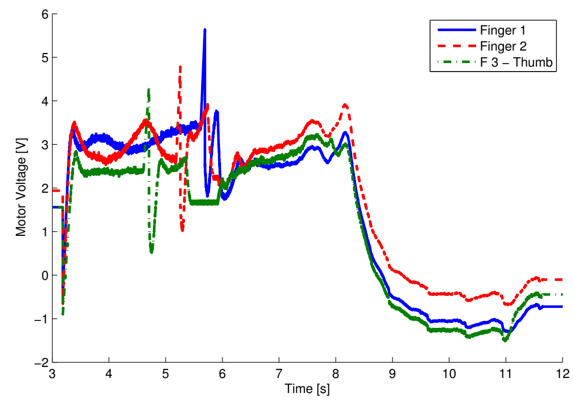
(a) Finger positions. $F = 10$ N.



(b) Tactile sensor measurements. $F = 10$ N.

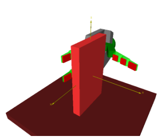


(c) Sum of tactile forces. $F = 10$ N.

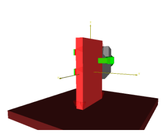


(d) Motor voltages. $F = 10$ N.

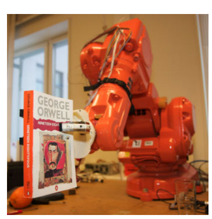
Figure 6: Finger positions and tactile forces for a real experiment under the same conditions as those in Figure 4. The final grasp force is set to 10 N. The grasp controller is engaged at $t = 3.18$ s and the fingers start to close. The hybrid controller is automatically activated at $t = 5.92$ s when all fingers have detected contact. The references are then increased during two seconds and then held.



(a) Initial position i simulation.



(b) Final grasp in simulation.



(c) Final grasp in reality after also lifting the book.

Figure 9: Grasping a book.

A. The KTHand

Developed primarily for wrapping power grasps, the KTHand cannot perform any fingertip precision grasps securely. Equipping it with tactile sensors on the fingertips, fingertip grasps could be performed. For small objects it is a problem that the active area of the sensor only covers the center of the finger and not the edges, causing some contacts to be undetected.

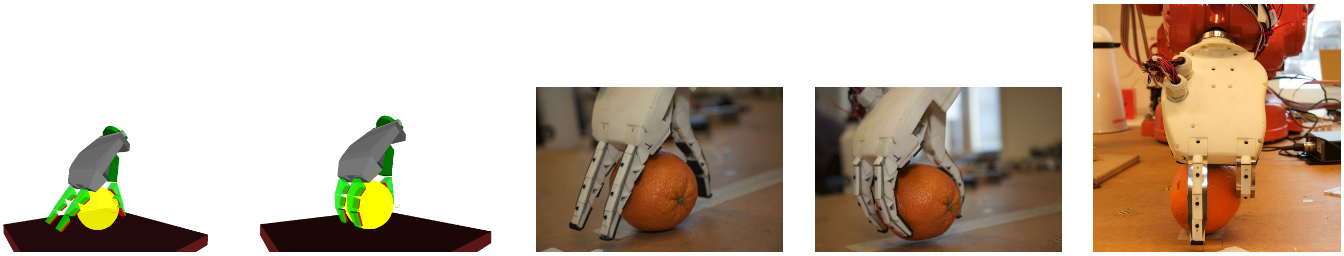
A project has just been started where the controller will be compiled to run on a dedicated microprocessor, relieving the need for a dedicated dSpace PC taking the KTHand project one step closer to the realization of a self-contained, autonomous robot hand.

B. The Simulation Environment

The major limitations in GraspIt! are with respect to the contact model implementation and joint friction. Implementing joint friction would make the simulation run much faster as a longer time step could be used. This would facilitate the iterative development process and allow building more simulated experience in a shorter time. If a high quality dynamic physical modeling is essential, for example when grasping very compliant objects such as sponges, towels et cetera, simulation tools using non-rigid objects are more suitable, see, e.g. [3], [7], [27].

C. Control

We know that the control parameters can be tuned further for even better performance. A a more elaborate controller, would



(a) Initial position i simulation. (b) Final grasp in simulation. (c) Initial positions in reality. (d) Successful grasp in reality. (e) Failed grasp in reality.

Figure 8: Grasping an orange.

benefit performance. A better estimation of the parameters of the hand would enable better models of actuators and joint friction and allow feed forward control.

IX. Conclusion

We have presented a system that includes an under-actuated anthropomorphic robot hand with control algorithms for autonomous grasping of everyday objects. The system can accomplish robust grasping of at least a few common shapes that fit the morphology of the hand by hybrid force/position control.

The performance of the robot hand and the grasp controller was evaluated and improved through a combination of dynamic simulation and real experiments. The control framework allows the same controller with only minor modifications to be used for simulation as well as real experiments. The results from comparing simulation results to real experiments suggest that simulation under certain conditions is a valid basis for generating grasp experience.

References

[1] A. Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road towards simplicity. *IEEE Trans. on Robotics and Automation*, 16(6):652–662, Dec 2000.

[2] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 348–353, April 2000.

[3] Tong Cui, Jing Xiao, and Aiguo Song. Simulation of grasping deformable objects with a virtual human hand. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3965–3970, September 2008.

[4] Noel Curtis and Jing Xiao. Efficient and effective grasping fo novel objects through learning and adapting a knowledge base. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2252–2257, September 2008.

[5] D. Ding, Y.-H. Liu, and S. Wang. Computing 3-d optimal formclosure grasps. In *In Proc. of the 2000 IEEE International Conference on Robotics and Automation*, pages 3573 – 3578, 2000.

[6] Aaron Edsinger. *Robot Manipulation in Human Environments*. PhD thesis, Massachusetts Inst. of Technology, 2007.

[7] G. Ferretti, G. Magnani, P. Rocco, and L. Viganò. Modelling and simulation of a gripper with dymola. *Mathematical and Computer Modelling of Dynamical Systems*, 2006.

[8] R. D. Howe. Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3):245–261, 1994.

[9] S. Larsson and J. A. P. Kjellander. Path planning for laser scanning with an industrial robot. *Robotics and Autonomous Systems*, 5(7):615–624, 2008.

[10] M. H. Lee and H. R. Nicholls. Tactile sensing for mechatronics - a state of the art survey. *Mechatronics*, 9(1):1–31, October 1999.

[11] Hong Liu, Peter Meusel, Gerd Hirzinger, Minghe Jin anc Yiwei Liu, and Zongwu Xie. The modular multisensory DLR-HIT-hand: Hardware and software architecture. *IEEE Trans. on Mechatronics*, 14(4), August 2008.

[12] A. T. Miller. *GraspIt!: A Versatile Simulator for Robotic Grasping*. PhD thesis, Department of Computer Science, Columbia University, 2001.

[13] A. T. Miller, S. Knoop, and Henrik I. Christensen P.K. Allen. Automatic grasp planning using shape primitives. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1824–1829, 2003.

[14] A. Morales, E. Chinellato, A. H. Fagg, and A.P. del Pobil. Using experience for assessing grasp reliability. *International Journal of Humanoid Robotics*, 1(4):671–691, 2004.

[15] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko. Development of a high-speed multifingered hand system and its application to catching. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 2666–2671, October 2003.

[16] R. J. Platt. *Learning and Generalizing Control-based Grasping and Manipulation Skills*. PhD thesis, University of Massachusetts, 2006.

[17] R. Platt Jr, A. H. Fagg, and R. A. Grupen. Extending fingertip grasping to whole body grasping. In *Proc. of the Intl. Conference on Robotics and Automation*, 2003.

[18] N. S. Pollard. *Parallel Methods for Synthesizing Whole-Hand Grasps from Generalized Prototypes*. PhD thesis, Massachusetts Inst. of Technology, 1994.

[19] N. S. Pollard. Closure and quality equivalence for efficient synthesis of grasps from examples. *International Journal of Robotic Research*, 23(6):595–613, 2004.

[20] A. Skoglund, B. Iliev, and R. Palm. A hand state approach to imitation with a next-state-planner for industrial manipulators. In *Int. Conf. on Cognitive Systems*, Karlsruhe, Germany, April 2008.

[21] J. Tegin, S. Ekvall, D. Kragic, J. Wikander, and B. Iliev. Demonstration based learning and control for automatic grasping. 2(1), January 2009.

[22] J. Tegin and J. Wikander. Tactile sensing in intelligent robotic manipulation – a review. *Industrial Robot*, 32(1):64–70, 2004.

[23] J. Tegin and J. Wikander. A framework for grasp simulation and control in domestic environments. In *IFAC-Symp. on Mechatronic Syst.*, Heidelberg, Germany, September 2006.

[24] J. Tegin, J. Wikander, and B. Iliev. A sub €1000 robot hand for grasping – design, simulation and evaluation. In *Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Coimbra, Portugal, September 2008.

[25] W. Townsend. The barretthand grasper – programmably flexible part handling and assembly. *Industrial Robot: An International Journal*, 27(3):181–188, 2000.

[26] R. Volpe and P. Khosla. A theorethical and experimental investigation of explicit force control strategies for manipulators. *IEEE Trans. Automatic Control*, 38(11):1634–1650, November 1993.

[27] Loredana Zollo, Stefano Roccella, Eugenio Guglielmelli, M. Chiara Carrozza, and Paulo Dario. Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *IEEE Trans. on Mechatronics*, 12(4), August 2008.